(19) JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: **10049368 A**

(43) Date of publication of application: **20.02.98**

(51) Int. Cl **G06F 9/32**

(21) Application number: **08200847**

(22) Date of filing: **30.07.96**

(71) Applicant: **MITSUBISHI ELECTRIC CORP**

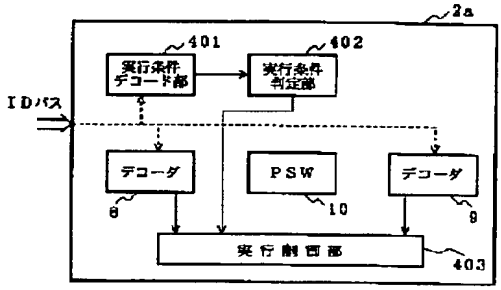
(72) Inventor: **YAMADA AKIRA**
YOSHIDA TOYOHIKO
KENGAKU TOORU

(54) **MICROPORCESSOR HAVING CONDITION EXECUTION INSTRUCTION**

(57) Abstract:

PROBLEM TO BE SOLVED: To decrease the number of bits in an execution condition field against the condition number by executing an instruction when the coincidence is confirmed between the decoding result of an execution condition code part and the condition that is set by a general-purpose flag.

SOLUTION: An execution condition decoding part 401 extracts an execution condition field of an instruction and decodes the value of extracted three bits into the data corresponding to an execution control flag. If the value of three bits are equal to '000', the relevant code shows that it always executes the instructions. Thereby, the data are outputted to show that the execution control flag is ignored. An execution condition decision part 402 compares these data with the execution control flag. In such cases, the execution control flag is ignored and therefore the part 402 instructs an execution control part 403 to execute the relevant instruction. In response to this instruction, the part 403 outputs a control signal to a memory unit or an integer arithmetic unit.



(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号

特開平10－49368

(43)公開日　平成10年(1998)２月20日

(51)Int.Cl.[6]	識別記号	庁内整理番号	ＦＩ		技術表示箇所
Ｇ０６Ｆ　9/32	３２０		Ｇ０６Ｆ　9/32	３２０Ｆ	

審査請求　未請求　請求項の数４　ＯＬ　（全 18 頁）

(21)出願番号　特願平8－200847

(22)出願日　平成８年(1996)７月30日

(71)出願人　000006013
三菱電機株式会社
東京都千代田区丸の内二丁目２番３号

(72)発明者　山田　朗
東京都千代田区丸の内二丁目２番３号　三菱電機株式会社内

(72)発明者　吉田　豊彦
東京都千代田区丸の内二丁目２番３号　三菱電機株式会社内

(72)発明者　見学　徹
東京都千代田区丸の内二丁目２番３号　三菱電機株式会社内

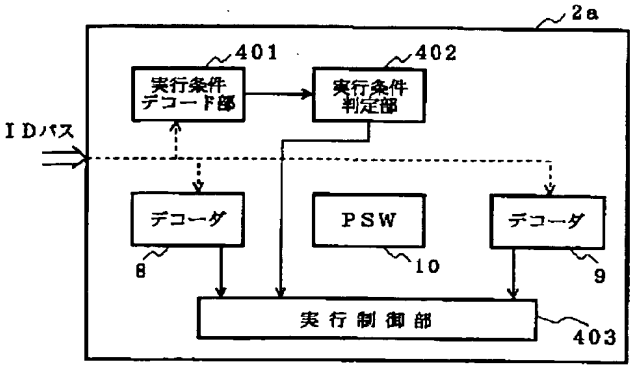
(74)代理人　弁理士　田澤　博昭　（外２名）

(54)【発明の名称】　条件実行命令を有するマイクロプロセッサ

(57)【要約】

【課題】　命令フォーマット中の実行条件フィールドのビット数が多い。

【解決手段】　命令における実行条件フィールドは、命令を実行することを示す汎用フラグの条件がエンコードされた値を有する。命令デコードユニット２は、実行条件フィールド１０５の値をデコードする実行条件デコード部４０１と、実行条件デコード部４０１のデコード結果と汎用フラグによる条件とが合致しているか否か判定し、合致していた場合に命令を実行することに決定する実行条件判定部４０２とを備える。



1

【特許請求の範囲】

【請求項１】　実行条件フィールドおよび演算フィールドを含む命令をデコードする命令デコーダと、命令の実行を制御する情報が設定されるレジスタであって条件が設定される汎用フラグを含む制御レジスタと、前記命令デコーダの出力に従って命令を実行する命令実行部とを備えた条件実行命令を有するマイクロプロセッサにおいて、前記実行条件フィールドは、命令を実行することを示す前記汎用フラグの条件がエンコードされた値を有するものであり、前記命令デコーダは、前記実行条件フィールドの値をデコードする実行条件デコード部と、前記実行条件デコード部のデコード結果と前記汎用フラグによる条件とが合致しているか否か判定し、合致していた場合に命令を実行することに決定する実行条件判定部とを備えたことを特徴とする条件実行命令を有するマイクロプロセッサ。

【請求項２】　実行条件フィールドは、条件実行の判定に用いられる汎用フラグの全ての組み合わせを表現するビット数よりも少ないビット長を有し、実行条件デコード部は、前記ビット長のエンコード値をデコードすることを特徴とする請求項１記載の条件実行命令を有するマイクロプロセッサ。

【請求項３】　実行条件フィールドおよび演算フィールドを含む命令をデコードする命令デコーダと、命令の実行を制御する情報が設定されるレジスタであって条件が設定される汎用フラグを含む制御レジスタと、データを記憶する汎用レジスタと、前記命令デコーダの出力に従って命令を実行する命令実行部とを備えた条件実行命令を有するマイクロプロセッサにおいて、前記実行条件フィールドは、命令を実行することを示す前記汎用フラグの条件が設定された前記汎用レジスタ中のレジスタを指定する値を有するものであり、前記命令デコーダは、前記実行条件フィールドで指定されたレジスタを参照するレジスタ参照部と、前記レジスタ参照部が参照したレジスタに設定されている条件と前記汎用フラグによる条件とが合致しているか否か判定し、合致していた場合に命令を実行することに決定する実行条件判定部とを備えたことを特徴とする条件実行命令を有するマイクロプロセッサ。

【請求項４】　命令デコーダは、１つの実行条件フィールド、１つの命令選択条件フィールドおよび複数の演算フィールドを含む命令をデコードし、命令実行部は、複数の演算回路を有し、前記命令デコーダは、さらに、前記命令選択条件フィールドの設定値に応じて前記複数の演算フィールドによる操作を行うか否か定める実行制御部を備えたことを特徴とする請求項１から請求項３のうちのいずれか１項記載の条件実行命令を有するマイクロプロセッサ。

【発明の詳細な説明】

【０００１】

2

【発明の属する技術分野】この発明は、命令の一部に含まれる実行条件フィールドを用いて命令の実行を制御する条件実行命令を有するマイクロプロセッサに関するものである。

【０００２】

【従来の技術】マイクロプロセッサにおけるパイプライン実行の分岐のペナルティを低減してマイクロプロセッサの性能を向上させるために、条件実行や投機実行などの手法がある。図１６は例えばＶＬＳＩテクノロジー社の「ＡＲＭ（ACORN RISC MACHINE）ファミリーデータマニュアル（１９９０年）」、２－２９頁に示された従来のＲＩＳＣマイクロプロセッサの命令フォーマットの一例を示す説明図である。図に示すように、このマイクロプロセッサにおける条件実行命令は、４ビットの実行条件フィールドを有する。マイクロプロセッサのネガティブフラグ（Ｎ）、ゼロフラグ（Ｚ）、キャリーフラグ（Ｃ）およびオーバフローフラグ（Ｖ）の各状態と実行条件フィールドの一致／不一致に応じて、命令が実行されるか否か決定される。

【０００３】しかし、条件実行の判定に用いられる各フラグは、特定の命令によってのみ設定される専用フラグである。そのために、このようなマイクロプロセッサにおいては、条件実行の範囲が制限される。

【０００４】図１７は特開平７－１８２１６５号公報に示された従来の他のマイクロプロセッサの命令フォーマット３００を示す説明図である。図において、３０１は実行条件が設定される実行条件フィールド、３０２は操作部分である演算フィールドである。実行条件フィールド３０１において、３０３，３０４はマイクロプロセッサの内部のフラグ＃０に関連するCv0ビット、Cd0ビット、３０５，３０６はマイクロプロセッサの内部のフラグ＃１に関連するCv1ビット、Cd1ビット、３０７，３０８はマイクロプロセッサの内部のフラグ＃２に関連するCv2ビット、Cd2ビットである。

【０００５】図１８はCv ビットおよびCd ビットとそれらが表す意味との関係を示す説明図である。図１８において、Cv ビットは、Cv0ビット３０３、Cv1ビット３０５およびCv2ビット３０７を代表している。Cd ビットは、Cd0ビット３０４、Cd1ビット３０６およびCd2ビット３０８を代表している。図１８に示すように、実行条件フィールド３０１におけるCv0ビット３０３およびCd0ビット３０４がともに「１」に設定されている場合には、フラグ＃０が真の状態にあるときに命令が実行されうる条件が成立する。Cv0ビット３０３が「１」に設定されていてCd0ビット３０４が「０」に設定されている場合には、フラグ＃０が偽の状態にあるときに命令が実行されうる条件が成立する。Cv0ビット３０３が「０」に設定されている場合には、フラグ＃０の状態がいずれであっても命令が実行されうる条件が成立する。

【０００６】Cv1ビット３０５およびCd1ビット３０６

とフラグ＃１とについても、図１８に示された関係によって、命令が実行されうる条件が成立しているかどうか判断される。Cv2ビット３０７およびCd2ビット３０８とフラグ＃２とについても、図１８に示された関係によって、命令が実行されうる条件が成立しているかどうか判断される。なお、Cv ビットの値に応じて、真または偽の判断がなされるのか、それとも判断がなされないかが決まるので、Cv ビットは、条件の有効性を判断するためのビットである。また、Cd ビットの値に応じて、真であるか偽であるかの判断がなされるので、Cd ビットは、値が判断されるビットである。

【０００７】次に動作について説明する。マイクロプロセッサは、命令フォーマット３００における実行条件フィールド３０１の設定状態に応じて、演算フィールド３０２が示す命令を実行するかどうか制御する。例えば、実行条件フィールド３０１の設定値が「１１１１１」であった場合には、マイクロプロセッサは、フラグ＃０，＃１，＃２が全て真であったときに、演算フィールド３０２が示す命令を実行する。

【０００８】フラグ＃０，＃１，＃２がそれぞれ１ビットで表されているときには、例えば、真は「１」に対応し、偽は「０」に対応する。特開平７－１８２１６５号公報に示されたマイクロプロセッサでは、フラグ＃０，＃１，＃２をそれぞれ２ビットで表し、真、偽および未定の３状態を表現する。例えば、フラグ＃０，＃１，＃２中に状態が未定のものがあった場合には、その他の条件が成立していれば、演算フィールド３０２が示す命令が実行される。そして、実行結果が、一般のレジスタファイルとは異なるシャドウレジスタファイルに書き込まれる。全ての条件が成立した場合には、その時点で、シャドウレジスタファイルの内容がレジスタファイルに書き込まれる。このようにして、投機実行が実現されている。また、特開平７－１８２１６５号公報には、条件数を増やすために、フラグ数がm個（m＞３）の場合も示されている。

【０００９】

【発明が解決しようとする課題】従来の条件実行命令を有するマイクロプロセッサは以上のように構成されているので、条件実行の範囲は拡大されるものの、例えば３つのフラグに対して６ビットの実行条件フィールド３０１が必要とされる。よって、命令フォーマット３００中の実行条件フィールド３０１のビット数が多いという課題があった。

【００１０】この発明は上記のような課題を解決するためになされたもので、条件数に対する実行条件フィールドのビット数を低減できる条件実行命令を有するマイクロプロセッサを得ることを目的とする。なお、特開平２－２２８７３号公報には、分岐命令やジャンプ命令を行うか否か判断するための条件のエンコード値がレジスタに設定され、分岐命令やジャンプ命令を行うか否か判断するための値としてデコード値がレジスタから出力されるものが開示されている。しかし、特開平２－２２８７３号公報には、命令中の実行条件フィールドの構成に関して何等言及されていない。

【００１１】

【課題を解決するための手段】請求項１記載の発明に係る条件実行命令を有するマイクロプロセッサは、命令における実行条件フィールドが、命令を実行することを示す汎用フラグの条件がエンコードされた値を有するものであり、命令デコーダが、実行条件フィールドの値をデコードする実行条件デコード部と、実行条件デコード部のデコード結果と汎用フラグによる条件とが合致しているか否か判定し合致していた場合に命令を実行することに決定する実行条件判定部とを備えたものである。

【００１２】請求項２記載の発明に係る条件実行命令を有するマイクロプロセッサは、実行条件フィールドが、条件実行の判定に用いられる汎用フラグの全ての組み合わせを表現するビット数よりも少ないビット長を有し、実行条件デコード部が、そのようなビット長のエンコード値をデコードする構成になっているものである。

【００１３】請求項３記載の発明に係る条件実行命令を有するマイクロプロセッサは、実行条件フィールドが、命令を実行することを示す汎用フラグの条件が設定された汎用レジスタ中のレジスタを指定する値を有するものであり、命令デコーダが、実行条件フィールドで指定されたレジスタを参照するレジスタ参照部と、レジスタ参照部が参照したレジスタに設定されている条件と汎用フラグによる条件とが合致しているか否か判定し合致していた場合に命令を実行することに決定する実行条件判定部とを備えたものである。

【００１４】請求項４記載の発明に係る条件実行命令を有するマイクロプロセッサは、命令実行部が、複数の演算回路を有し、命令デコーダが、実行条件フィールド、命令選択条件フィールドおよび複数の演算フィールドを含む命令をデコードする構成であって、さらに、命令選択条件フィールドの設定値に応じて複数の演算フィールドによる操作を行うか否か定める実行制御部を有する構成になっているものである。

【００１５】

【発明の実施の形態】以下、この発明の実施の一形態を説明する。

実施の形態１．図１はこの発明の実施の一形態によるマイクロプロセッサの構成を示すブロック図である。このマイクロプロセッサは、３２ビットの内部データバスを有する３２ビットマイクロプロセッサである。図において、２は命令ＲＡＭ６から６４ビット幅のＩＤバスを介して入力した命令コードをデコードする処理を行う命令デコードユニット（命令デコーダ）、３はアドレス計算を行うメモリユニット（命令実行部）、４は論理演算やシフト演算を行う整数演算ユニット（命令実行部）、５

は３２ビット×６４ワードの汎用レジスタ、７はデータが格納されるデータＲＡＭである。

【００１６】命令デコードユニット２において、８，９はそれぞれ命令コードをデコードするデコーダ、１０はプロセッサの状態を示すプロセッサ状態語（Processor Status Word ；以下、プロセッサ状態語をＰＳＷという）である。命令デコードユニット２は、さらに、デコーダ８のデコード結果とＰＳＷ１０の内容にもとづいて制御信号１１を作成し、それをメモリユニット３に与える。また、命令デコードユニット２は、デコーダ９のデコード結果とＰＳＷ１０の内容にもとづいて制御信号１２を作成し、それを整数演算ユニット４に与える。

【００１７】メモリユニット３において、１３はジャンプや分岐を含まない命令を実行するとＰＣ（プログラムカウンタ）値に８を加えて次に実行する命令に対するＰＣ値を算出するとともに、ジャンプや分岐を含む命令の実行時に分岐変位をＰＣ値に加算したり、演算で指定されたアドレッシングモードに応じた計算を行ってジャンプ先の命令に対するＰＣ値を計算するＰＣ制御部である。また、ＰＣ制御部１３は、計算したＰＣ値を３２ビット幅のＩＡバスを介して命令ＲＡＭ６に送り、命令ＲＡＭ６から命令コードを出力させる。１４はオペランドとなるデータのアクセスを制御するメモリ制御部である。メモリ制御部１４は、３２ビット幅のＤＡバスを介してアドレスデータをデータＲＡＭ７に転送し命令実行に必要なデータを６４ビット幅のＤＤバスを介してアクセスする。１５は汎用レジスタ５から３２ビット幅のＳ１バス、Ｓ２バス、Ｓ３バスを介して転送された最大３ワードのデータを用いて算術論理演算を行い演算結果を３２ビット幅のＤ１バスを介して汎用レジスタ５に転送するＡＬＵ、１６は汎用レジスタ５からＳ１バス、Ｓ２バス、Ｓ３バスを介して転送されたデータを用いてシフト演算を行い演算結果をＤ１バスを介して汎用レジスタ５に転送するシフタである。

【００１８】メモリユニット３に対して、Ｓ１バス、Ｓ２バス、Ｓ３バス、Ｓ４バスを介して、３２ビット長のデータを一時に４ワード転送することが可能である。従って、例えば、第１のレジスタの内容と第２のレジスタの内容との和でアドレッシングされるメモリの領域に第３のレジスタの内容をストアするとともに、第３のレジスタの内容をストアしたアドレスに所定値を加算して得られる値でアドレッシングされるメモリの領域に第４のレジスタの内容をストアする２ワードストア命令を実現することができる。また、メモリユニット３は、Ｄ１バスおよびＤ２バスを介して、メモリユニット３内での２ワードの演算結果またはデータＲＡＭ７から転送された２ワードのデータを汎用レジスタ５に転送することができる。

【００１９】整数演算ユニット４において、１７は汎用レジスタ５から３２ビット幅のＳ４バス、Ｓ５バス、Ｓ６バスを介して転送された最大３ワードのデータを用いて乗算を行い演算結果を３２ビット幅のＤ２バス、Ｄ３バスを介して汎用レジスタ５に転送する乗算器、１８は乗算の結果を累積加算または累積減算して保持するアキュムレータである。アキュムレータとして、６４ビットのものが２本ある。１９は汎用レジスタ５からＳ４バス、Ｓ５バス、Ｓ６バスを介して転送された最大３ワードのデータを用いて算術論理演算を行い演算結果をＤ２バス、Ｄ３バスを介して汎用レジスタ５に転送するＡＬＵ、２０は汎用レジスタ５からＳ４バス、Ｓ５バス、Ｓ６バスを介して転送されたデータを用いてシフト演算を行い演算結果をＤ２バス、Ｄ３バスを介して汎用レジスタ５に転送するシフタである。

【００２０】このマイクロプロセッサでは、汎用レジスタ５から、最大６種類のレジスタ値を読み出すことが可能であって、読み出されたデータは、それぞれ、Ｓ１バス、Ｓ２バス、Ｓ３バス、Ｓ４バス、Ｓ５バス、Ｓ６バスに出力される。また、汎用レジスタ５には、Ｄ１バス、Ｄ２バス、Ｄ３バスを介して最大３種類のレジスタ値を同時に書き込むことが可能である。

【００２１】図２はこのマイクロプロセッサの命令フォーマットを示す説明図である。命令フォーマットとして、（ａ）に示すような１つの命令コードで２つの演算（operation）を指示する２演算命令のフォーマット１０１と、（ｂ）に示すような１つの命令コードで１つの演算を指示する１演算命令のフォーマット１０２とがある。２演算命令のフォーマット１０１には、フィールド１０３およびフィールド１０４からなるフォーマットフィールドと、２つの演算フィールド１０６，１０７と、各演算フィールド１０６，１０７に付属する各実行条件フィールド１０５とがある。１演算命令のフォーマット１０２には、フィールド１０３およびフィールド１０４からなるフォーマットフィールドと、フィールド１０８，１０９，１１０からなる演算フィールドと、演算フィールドに付属する実行条件フィールド１０５とがある。

【００２２】フォーマットフィールドは、以下のような意味を示す。

コード：フォーマット	実行の順番 operation_0	operation_1
ＦＭ＝００：２命令	第１	第１
０１：２命令	第１	第２
１０：２命令	第２	第１
１１：１命令	第１	……

ここで、ＦＭは、フィールド１０３およびフィールド１０４からなる２ビットの値である。

【００２３】ＦＭ＝００の場合、この命令は２演算命令であることを示す。そして、演算フィールド１０６で指示されたoperation_０の演算と演算フィールド１０７で指示されたoperation_１の演算とが、デコード直後のクロックサイクルで並列に実行される。operation_０の演算はメモリユニット３で実行され、operation_１の演算は整数演算ユニット４で実行される。ＦＭ＝０１の場合、この命令は２演算命令であることを示す。そして、operation_０の演算が、デコード直後のクロックサイクルで実行され、operation_１の演算が、operation_０の演算に対して、１クロックサイクル遅れて実行される。ＦＭ＝１０の場合、この命令は２演算命令であることを示す。そして、operation_１の演算が、デコード直後のクロックサイクルで実行され、operation_０の演算が、operation_１の演算に対して、１クロックサイクル遅れて実行される。ＦＭ＝１１の場合、この命令は１演算命令であることを示す。そして、フィールド１０８，１０９，１１０からなる演算フィールドで指示された１つの演算がデコード直後のクロックサイクルで実行される。

【００２４】実行条件フィールド１０５は、以下のような意味を持つ。

コード：	実行条件		
ＣＣ＝０００：	常時		
００１：	Ｆ０＝真	かつ	Ｆ１＝無視
０１０：	Ｆ０＝偽	かつ	Ｆ１＝無視
０１１：	Ｆ０＝無視	かつ	Ｆ１＝真
１００：	Ｆ０＝無視	かつ	Ｆ１＝偽
１０１：	Ｆ０＝真	かつ	Ｆ１＝真
１１０：	Ｆ０＝真	かつ	Ｆ１＝偽
１１１：	予約済		

【００２５】実行条件フィールド１０５は、マイクロプロセッサの実行コントロールフラグＦ０，Ｆ１の値に応じて、演算フィールド１０６，１０７のoperation_０の演算やoperation_１の演算、およびフィールド１０８，１０９，１１０で構成される演算フィールドの演算が有効であるか無効であるか定める。実行コントロールフラグＦ０，Ｆ１については後で説明する。演算が有効であるとは、演算結果がレジスタ、メモリおよびフラグに反映され、その演算による動作の結果が残ることを意味する。演算が無効であるとは、演算結果がレジスタ、メモリおよびフラグに反映されず、あたかも無効演算（ＮＯＰ）が実行されたかのような動作の結果が残ることを意味する。

【００２６】実行条件フィールド１０５の値ＣＣ＝０００のときには、実行コントロールフラグＦ０，Ｆ１の値にかかわらず常に演算は有効である。ＣＣ＝００１のときには、実行コントロールフラグＦ０＝真のときにのみ演算は有効である。実行コントロールフラグＦ１の状態はいずれでもよい。ＣＣ＝０１０のときには、実行コントロールフラグＦ０＝偽のときにのみ演算は有効である。実行コントロールフラグＦ１の状態はいずれでもよい。ＣＣ＝０１１のときには、実行コントロールフラグＦ１＝真のときにのみ演算は有効である。実行コントロールフラグＦ０の状態はいずれでもよい。ＣＣ＝１００のときには、実行コントロールフラグＦ１＝偽のときにのみ演算は有効である。実行コントロールフラグＦ０の状態はいずれでもよい。ＣＣ＝１０１のときには、実行コントロールフラグＦ０＝真かつＦ１＝真のときにのみ演算は有効である。ＣＣ＝１１０のときには、実行コントロールフラグＦ０＝真かつＦ１＝偽のときにのみ演算は有効である。ＣＣ＝１１１のときの動作は未定義であり、ユーザは、ＣＣ＝１１１となる命令を用いることはできない。

【００２７】図３は演算フィールドの詳細な内容を示す説明図である。フォーマット１１１～１１７は、それぞれ２８ビットで表現される短型の演算フィールド１０６または演算フィールド１０７によるものである。フォーマット１１８，１１９は、フィールド１０８，１０９，１１０で構成される長型の演算フィールドによるものである。

【００２８】フォーマット１１１（Ｓｈｏｒｔ＿Ｍ）は、演算内容を指定するフィールド１２０、レジスタ番号を指定する２つのフィールド１２１，１２２、レジスタ番号または６ビット長の即値を指定するフィールド１２３、およびフィールド１２３がレジスタ番号を示すのか即値を示すのかを指定するフィールド１２４で構成される。図３に示すように、フィールド１２４の値Ｘが「００」、「０１」または「１１」であるときにはフィールド１２３がレジスタ番号を示していることを示し、「１０」であるときには即値を示していることを示す。このフォーマット１１１は、レジスタ間接アドレッシングのメモリアクセス演算に用いられる。

【００２９】フォーマット１１２（Ｓｈｏｒｔ＿Ａ）は、演算内容を指定するフィールド１２０、レジスタ番号を指定する２つのフィールド１２１，１２２、レジスタ番号または６ビット長の即値を指定するフィールド１２３、およびフィールド１２３がレジスタ番号を示すのか即値を示すのかを指定するフィールド１２５で構成される。図３に示すように、フィールド１２５の値Ｘ’が「０」であるときにはフィールド１２３がレジスタ番号を示していることを示し、「１」であるときには即値を示していることを示す。このフォーマット１１２は、算術演算、論理演算、シフト演算およびビット演算に用いられる。

【００３０】フォーマット１１３（Ｓｈｏｒｔ＿Ｂ１）は、演算内容を指定するフィールド１２０およびレジスタ番号を指定するフィールド１２６で構成される。このフォーマット１１３は、レジスタ指定によるジャンプ命

令および分岐命令に用いられる。フォーマット１１４（Ｓｈｏｒｔ＿Ｂ２）は、演算内容を指定するフィールド１２０および１８ビット長のディスプレイスメントのフィールド１２７で構成される。このフォーマット１１４は、ジャンプ命令および分岐命令に用いられる。

【００３１】フォーマット１１５（Ｓｈｏｒｔ＿Ｂ３）は、演算内容を指定するフィールド１２０、レジスタ番号を指定するフィールド１２１、レジスタ番号または１２ビット長の即値を指定するフィールド１２８、フィールド１２８がレジスタ番号を示すのか即値を示すのかを指定するフィールド１２９、およびゼロ判定にもとづいてフィールド１２１にもとづく条件ジャンプまたは条件分岐を行うか否か指定するフィールド１３０で構成される。このフォーマット１１５は、条件ジャンプ命令および条件分岐命令に使用される。

【００３２】フォーマット１１６（Ｓｈｏｒｔ＿Ｄ１）は、演算内容を指定するフィールド１２０、レジスタ番号を指定するフィールド１２１、レジスタ番号または１２ビット長の即値を指定するフィールド１２８、フィールド１２８がレジスタ番号を示すのか即値を示すのかを指定するフィールド１２９で構成される。このフォーマット１１６は、条件ジャンプ命令、条件分岐命令およびリピート命令に使用される。フォーマット１１７（Ｓｈｏｒｔ＿Ｄ２）は、演算内容を指定するフィールド１２０、レジスタ番号または１２ビット長の即値を指定するフィールド１２８、フィールド１２８がレジスタ番号を示すのか即値を示すのかを指定するフィールド１２９、遅延命令（ディレイド命令）に関するフィールド１３１で構成される。このフォーマット１１７は、ディレイドジャンプ命令、ディレイド分岐命令およびリピート命令に使用される。

【００３３】フォーマット１１８（Ｌｏｎｇ１）は、演算内容を指定するフィールド１２０、レジスタ番号を指定する２つのフィールド１２１，１２２、３２ビット長の即値を指定するフィールド１３２で構成される。このフォーマット１１８は、複雑な算術演算、大きな即値を用いる算術演算、大きなディスプレイスメント付きレジスタ間接アドレッシングのメモリアクセス演算、大きな変位の分岐演算および絶対番地へのジャンプ命令などに使用される。フォーマット１１９（Ｌｏｎｇ２）は、演算内容を指定するフィールド１２０、レジスタ番号を指定する２つのフィールド１２１，１２２、３２ビット長の即値を指定するフィールド１３２、およびゼロ判定にもとづいてフィールド１３２にもとづく条件ジャンプまたは条件分岐を行うか否か指定するフィールド１３３で構成される。このフォーマット１１９は、大きな分岐変位を持つ条件ジャンプや条件分岐命令に使用される。

【００３４】図４はマイクロプロセッサのレジスタ構成を示す説明図である。このマイクロプロセッサは、図４（ａ）に示すような６４本の３２ビット長の汎用レジスタ５、図４（ｂ）に示すような１２本の制御レジスタ１５０、および図４（ｃ）に示すような２本のアキュムレータ１８を持つ。Ｒ０の汎用レジスタ１４０の内容は常に０であり、そこへの書き込みは無視される。Ｒ６２の汎用レジスタは、サブルーチンからの戻り先アドレスが設定されるリンクレジスタである。Ｒ６３の汎用レジスタは、スタックポインタであり、ＰＳＷ１０のＳＭフィールドの値に応じてユーザスタックポインタ（ＳＰＵ）または割り込みスタックポインタ（ＳＰＩ）として動作する。制御レジスタ１５０には、プログラムカウンタ１５１、ＰＳＷ１０、および各種の専用レジスタが含まれる。図３に示すフォーマット１１２による演算では、６４本の汎用レジスタ５のそれぞれを上位１６ビットと下位１６ビットとに分けてアクセスできる。また、２本のアキュムレータ１８を、上位３２ビットと下位３２ビットとに分けて別々にアクセスできる。

【００３５】図５はＰＳＷ１０の詳細内容を示す説明図である。ＰＳＷ１０の上位１６ビットには、スタックポインタを切り替えるためのＳＭフィールド１７１、セルフデバッグトラップ（ＳＤＢＴ）の検出を示すＥＡフィールド１７２、ＳＤＢＴの許可を指定するＤＢフィールド１７３、割り込み許可を指定するＩＥフィールド１７４、リピート動作の許可を指定するＲＰフィールド１７５、モジュロアドレッシングの許可を指定するＭＤフィールド１７６がある。下位１６ビットはフラグフィールド１８０である。フラグフィールド１８０には８個のフラグがあり、その中のＦ０フラグ１８１およびＦ１フラグ１８２は演算の有効／無効を指定する。各フラグの値は比較演算や算術演算の結果に応じて変化する。また、フラグ初期化演算で初期化したり、フラグ値書き込み演算で任意の値をフラグフィールド１８０に書き込むことによって変化する。フラグフィールド１８０の内容は、フラグ値読み出し演算によって読み出される。

【００３６】各フラグは、以下のような意味を有する。

ＳＭ＝０　：スタックモード０→ＳＰＩを使用
ＳＭ＝１　：スタックモード１→ＳＰＵを使用
ＥＡ＝０　：ＳＤＢＴを未検出
ＥＡ＝１　：ＳＤＢＴを検出
ＤＢ＝０　：ＳＤＢＴを非許可
ＤＢ＝１　：ＳＤＢＴを許可
ＩＥ＝０　：割り込み非許可
ＩＥ＝１　：割り込み許可
ＲＰ＝０　：リピートブロック無効
ＲＰ＝１　：リピートブロック有効
ＭＤ＝０　：モジュロアドレッシング無効
ＭＤ＝１　：モジュロアドレッシング有効
Ｆ０　：汎用フラグ（実行コントロールフラグ）
Ｆ１　：汎用フラグ（実行コントロールフラグ）
Ｆ２　：汎用フラグ
Ｆ３　：汎用フラグ

11

F４（S）：飽和演算フラグ
F５（V）：オーバーフローフラグ
F６（VA）：累積オーバーフローフラグ
F７（C）：キャリー／ボローフラグ

A－１．ロード／ストア命令

LDB : Load one byte to a register with sign extension
［１バイトロード（符号拡張あり）］
LDBU : Load one byte to a register with zero extension
［１バイトロード（ゼロ拡張あり）］
LDH : Load one half-word to a register with sign extension
［１ハーフワードロード（符号拡張あり）］
LDHH : Load one half-word to a register high
［１ハーフワードロード（レジスタ上位へ）］
LDHU : Load one half-word to a register with zero extension
［１ハーフワードロード（ゼロ拡張あり）］
LDW : Load one word to a register
［１ワードロード］
LD２W : Load two words to registers
［２ワードロード］
LD４BH : Load four bytes to four half-words in two registers with sign extension
［４バイトロード（２レジスタへ、符号拡張あり）］
LD４BHU : Load four bytes to four half-words in two registers with zero extension
［４バイトロード（２レジスタへ、ゼロ拡張あり）］
LD２H : Load two half-words to two words in two registers with sign extension
［２ハーフワードロード（２レジスタへ、符号拡張あり）］
STB : Store one byte from a register
［１バイトストア］
STH : Store one half-word from a register
［１ハーフワードストア］
STHH : Store one half-word from a register high
［１ハーフワードストア（レジスタ上位から）］
STW : Store one word from a register
［１ワードストア］
ST２W : Store two words from registers
［２ワードストア］
ST４HB : Store four bytes from four half-words from two registrers
［４バイトストア（２レジスタの４ハーフワードから）］
ST２H : Store two half-words from two registrers
［２ハーフワードストア（２レジスタから）］
MODDEC : Decrement a register value by a 5-bits immediate value
［即値５ビットのデクリメント］
MODINC : Increment a register value by a 5-bits immediate value
［即値５ビットのインクリメント］

12

【００３７】以下、このマイクロプロセッサの命令一覧を示す。

A．マイクロプロセッサ機能に関する命令

【００３８】

A－２．転送命令

MVFSYS : Move a control register to a general purpose register

[制御レジスタから汎用レジスタへ]

MVTSYS : Move a general purpose register to a control register
[汎用レジスタから制御レジスタへ]

MVFACC : Move a word from an accumulator
[アキュムレータからの１ワード転送]

MVTACC : Move two general purpose registers to an accumulator
[２汎用レジスタ内容のアキュムレータへの転送]

【００３９】

Ａ－３．比較命令

CMPcc : Compare [比較]
cc＝ＥＱ（等しい），ＮＥ（不等），ＧＴ（より大），
ＧＥ（以上），ＬＴ（未満），ＬＥ（以下），
ＰＳ（ともに正），ＮＧ（ともに負）

CMPUcc : Compare unsigned [比較（符号なし）]
cc＝ＧＴ，ＧＥ，ＬＴ，ＬＥ

【００４０】Ａ－４．最大値／最小値命令

【００４１】

reserved [予約済]

Ａ－５．算術演算命令

ＡＢＳ : Absolute [絶対値をとる]

ＡＤＤ : Add [加算]

ＡＤＤＣ : Add with carry [加算（キャリー付き）]

ＡＤＤＨｐｐｐ : Add half-word [ハーフワード加算]
ｐｐｐ＝ＬＬＬ（レジスタ下位、レジスタ下位、レジスタ下位），ＬＬＨ（レジスタ下位、レジスタ下位、レジスタ上位），ＬＨＬ，ＬＨＨ，ＨＬＬ，ＨＬＨ，ＨＨＬ，ＨＨＨ

ＡＤＤＳ : Add register Rb with the sign of the third operand
[レジスタＲｂに符号を付ける]

ＡＤＤＳ２Ｈ : Add sign to two half-words
[２ハーフワードに符号を付ける]

ＡＤＤ２Ｈ : Add two pairs of half-words
[２ハーフワード同士の加算]

ＡＶＧ : Average with rounding towards positive infinity
[平均をとる]

ＡＶＧ２Ｈ : Average two pairs of half-words rounding towards positive infinity
[２ハーフワードそれぞれの平均をとる]

ＪＯＩＮｐｐ : Join two half-words [２ハーフワードの結合]
ｐｐ＝ＬＬ，ＬＨ，ＨＬ，ＨＨ

ＳＵＢ : Subtract [減算]

ＳＵＢＢ : Subtract with borrow [ボロー付き減算]

ＳＵＢＨｐｐｐ : Subtract half-word [ハーフワードの減算]
ｐｐｐ＝ＬＬＬ，ＬＬＨ，ＬＨＬ，ＬＨＨ，ＨＬＬ，ＨＬＨ，ＨＨＬ，ＨＨＨ

ＳＵＢ２Ｈ : Subtract two pairs of half-words
[２つのハーフワードの減算]

【００４２】

Ａ－６．論理演算命令

ＡＮＤ : logical AND [論理積]

ＯＲ : logical OR [論理和]

15 16

NOT : logical NOT [反転]
XOR : logical exclusive OR [排他的論理和]
ANDFG : logical AND flags [フラグの論理積]
ORFG : logical OR flags [フラグの論理和]
NOTFG : logical NOT a flag [フラグの反転]
XORFG : logical exclusive OR flags [フラグの排他的論理和]

【００４３】

A－７．シフト演算命令
SRA : Shift right arithmetic [算術右シフト]
SRA2H : Shift right arithmetic two half-words [２つのハーフワードの算術右シフト]
SRC : Shift right concatenated registers [レジスタ連鎖右シフト]
SRL : Shift right logical [論理右シフト]
SRL2H : Shift right logical two half-words [２つのハーフワードの論理右シフト]
ROT : Rotate right [右回転]
ROT2H : Rotate right two half-words [２つのハーフワードの右回転]

【００４４】A－８．ビット操作命令
BCLR : Clear a bit [ビットクリア]
BNOT : Invert a bit [ビット反転]
BSET : Set a bit [ビットセット]
BTST : Test a bit [ビットテスト]

【００４５】

A－９．分岐命令
BRA : Branch [分岐]
BRATZR : Branch if zero [ゼロなら分岐]
BRATNZ : Branch if not zero [ゼロでないなら分岐]
BSR : Branch to subroutine [サブルーチンへ分岐]
BSRTZR : Branch to subroutine if zero [ゼロならサブルーチンへ分岐]
BSRTNZ : Branch to subroutine if not zero [ゼロでないならサブルーチンへ分岐]
JMP : Jump [無条件ジャンプ]
JMPTZR : Jump if zero [ゼロならジャンプ]
JMPTNZ : Jump if not zero [ゼロでないならジャンプ]
JSR : Jump to subroutine [サブルーチンへジャンプ]
JSRTZR : Jump to subroutine if zero [ゼロならサブルーチンへジャンプ]
JSRTNZ : Jump to subroutine if not zero [ゼロでないならサブルーチンへジャンプ]
NOP : No Operation [無操作]

[ディレイド分岐、ジャンプ命令に関する命令]
DBRA
DBRAI
DBSR
DBSRI
DJMP
DJMPI
DJSR
DJSRI

【００４６】

A－１０．ＯＳ関連命令
TRAP : Trap [トラップ]
REIT : Return from exception, interrupts and traps [例外、割り込み、トラップからのリターン]

【００４７】Ｂ．ＤＳＰ機能に関する命令

Ｂ－１．算術操作命令

ＭＵＬ　：Multiply [乗算]
ＭＵＬＸ　：Multiply with extended precision [倍精度乗算]
ＭＵＬＸＳ　：Multiply and shift to the right by one with extended precision
[倍精度乗算および１ビット右シフト]
ＭＵＬＸ２Ｈ　：Multiply two pairs of half-words with extended precision
[２ハーフワードずつの倍精度乗算]
ＭＵＬＨＸｐｐ：Multiply two half-words with extended precision
ｐｐ＝ＬＬ，ＬＨ，ＨＬ，ＨＨ
[２ハーフワードの倍精度乗算]
ＭＵＬ２Ｈ　：Multiply two pairs of half-words
[２ハーフワードずつの乗算]
ＭＡＣａ　：Multiply and add [積和演算]
ａ（アキュムレータ指定）＝０，１
ＭＡＣＳａ　：Multiply, shift to the right by one and add
ａ＝０，１
[１ビット右シフト付き積和演算]
ＭＳＵＢａ　：Multiply and subtract [積和（減算）演算]
ａ＝０，１
ＭＳＵＢＳａ　：Multiply, shift to the right by one and subtract
ａ＝０，１
[１ビット右シフト付き積和（減算）演算]

[飽和演算に関する命令]

ＳＡＴ
ＳＡＴＨＨ
ＳＡＴＨＬ
ＳＡＴＺ
ＳＡＴＺ２Ｈ
ＳＡＴ２Ｈ

【００４８】

Ｂ－２．リピート命令

ＲＥＰＥＡＴ　：Repeat a block of instructions
[命令ブロックの繰り返し]
ＲＥＰＥＡＴＩ：Repeat a block of instructions immediate
[命令ブロックの繰り返し（即値指定）]

【００４９】図６はマイクロプロセッサの並列２命令実行時のパイプライン動作を示す説明図である。この動作は、命令のフォーマットフィールドの値ＦＭ＝００のときに実行される。パイプライン１９０，１９５は、命令フェッチステージ１９１、デコード／アドレス演算ステージ１９２、実行／メモリアクセスステージ１９３、およびライトバックステージ１９４で構成される。並列２命令実行時には、メモリユニット３での実行と整数演算ユニット４での実行とが並列に行われる。図７はマイクロプロセッサのシーケンシャル命令実行時のパイプライン動作を示す説明図である。この動作は、命令のフォーマットフィールドの値ＦＭ＝０１，１０，１１のときに実行される。パイプライン２００は、命令フェッチステージ、デコード／アドレス演算ステージ、実行／メモリアクセスステージ、およびライトバックステージで構成されるが、この場合には、メモリユニット３での実行と整数演算ユニット４での実行とのうちのいずれかが、一時に実行される。

【００５０】図８は条件実行を用いるプログラムの一例を示す説明図である。図において、命令ｉ１，ｉ２による命令群Ｂ０は、フラグの状態によらず常に実行される。命令ｉ１によって、レジスタｒ１の値に「１」加算した値がレジスタｒ３に格納される。そして、命令ｉ２によって、レジスタｒ３とｒ４の内容が比較され、一致していれば実行コントロールフラグＦ０がセットされる。命令ｉ３，ｉ４による命令群Ｂ１は、実行コントロールフラグＦ０＝１のときに実行される。命令群Ｂ１が実行されると、命令ｉ３によって、レジスタｒ３の値とレジスタｒ１０の値との和が示すメモリアドレスから１ワードのデータがレジスタｒ６にロードされる。命令ｉ４によって、レジスタｒ５とｒ６の内容が比較され、一致していれば実行コントロールフラグＦ１がセットされ

る。命令ｉ６，ｉ７による命令群Ｂ３は、実行コントロールフラグＦ０＝１かつＦ１＝１のときに実行される。

【００５１】命令ｉ５による命令群Ｂ２は、実行コントロールフラグＦ０＝０のときに実行される。また、命令ｉ８による命令群Ｂ４は、実行コントロールフラグＦ０＝１かつＦ１＝０のときに実行される。命令ｉ１～ｉ８が順次記述されたプログラムがあった場合に、実行コントロールフラグＦ０，Ｆ１の値に応じて、図８において矢印で示すようにプログラムが実行される。

【００５２】図９は命令デコードユニット２ａにおいて条件実行を行うための構成を示すブロック図である。図において、４０１は命令における実行条件フィールド１０５をデコードする実行条件デコード部、４０２はデコードされた値ＣＣと実行コントロールフラグＦ０，Ｆ１とを比較する実行条件判定部、４０３は実行条件判定部４０２の比較結果に応じて命令の実行を制御する実行制御部である。その他のものは、図１に示すものと同じものである。

【００５３】次に動作について説明する。図８に示す命令ｉ１，ｉ２において、命令の実行条件フィールド１０５には、「０００」が設定されている。実行条件デコード部４０１は、命令における実行条件フィールド１０５を抽出する。そして、抽出した３ビットの値ＣＣを、例えば、実行コントロールフラグＦ０に対応するデータと実行コントロールフラグＦ１に対応するデータとにデコードする。３ビットの値ＣＣが「０００」であった場合には、既に説明したように、そのコードは命令を常時実行することを示している。そこで、実行コントロールフラグＦ０を無視することを示すデータと実行コントロールフラグＦ１を無視することを示すデータとを出力する。実行条件判定部４０２は、それらのデータと実行コントロールフラグＦ０，Ｆ１とを比較する。この場合には、実行コントロールフラグＦ０，Ｆ１は無視されるので、実行条件判定部４０２は、実行制御部４０３に対して、その命令を実行するように指示を出す。実行制御部４０３は、その指示に応じて、制御信号１１，１２をメモリユニット３または整数演算ユニット４に出力する。

【００５４】例えば、命令群Ｂ３の命令ｉ６，ｉ７において、実行条件フィールド１０５には、「１０１」が設定されている。実行条件デコード部４０１は、３ビットの値ＣＣを実行コントロールフラグＦ０に対応するデータと実行コントロールフラグＦ１に対応するデータとにデコードする。既に説明したように、そのコードは実行コントロールフラグＦ０＝真で実行コントロールフラグＦ１＝真のときに命令を実行することを示している。ここでは、真＝１とする。実行条件デコード部４０１は、実行条件として、実行コントロールフラグＦ０＝１を示すデータと実行コントロールフラグＦ１＝１を示すデータとを出力する。実行条件判定部４０２は、それらのデータと実行コントロールフラグＦ０，Ｆ１とを比較する。そして、実際に実行コントロールフラグＦ０＝１かつ実行コントロールフラグＦ１＝１であったときには、実行制御部４０３に対して、その命令を実行するように指示を出す。

【００５５】図１０は実行コントロールフラグをＦ０，Ｆ１，Ｆ２の３個とした場合の実行条件を示す説明図である。図に示すように、3^3 個の条件が考えられうる。全ての条件の指定を可能にするには、デコードされた値ＣＣとして５ビット必要である。すなわち、命令における実行条件フィールド１０５は５ビット長となる。例えば、ＣＣ＝「００００」の場合には、実行コントロールフラグＦ０＝真、Ｆ１＝真、Ｆ２＝真のときにその命令が実行される。

【００５６】この場合にも、実行条件デコード部４０１は、命令における実行条件フィールド１０５を抽出する。そして、抽出した５ビットの値ＣＣを、例えば、実行コントロールフラグＦ０に対応するデータ、実行コントロールフラグＦ１に対応するデータおよび実行コントロールフラグＦ２に対応するデータにデコードする。実行条件判定部４０２は、それらのデータと実行コントロールフラグＦ０，Ｆ１，Ｆ２とを比較し、それらのデータと実際の実行コントロールフラグＦ０，Ｆ１，Ｆ２の状態とが一致したときに、実行制御部４０３に対して、その命令を実行するように指示を出す。

【００５７】以上のように、実行条件フィールド１０５は、条件実行のための条件をエンコードして保持している。従って、３つの実行コントロールフラグＦ０，Ｆ１，Ｆ２のすべての組み合わせに対応した条件を５ビットの実行条件フィールド１０５で指定できる。図１７に示す従来の例では、3^3 個の条件を指定するのに６ビットの実行条件フィールド３０１が必要であったが、この実施の形態によれば、１ビット少ない実行条件フィールド１０５で同様の条件実行を実現できる。よって、演算フィールドに割り当てられるビット数を増加でき、マイクロプロセッサで提供できる命令数を増やすことができる。

【００５８】実行コントロールフラグＦ０，Ｆ１の数が２個の場合に、条件（真、偽、無視）の組み合わせは実は $3^2=9$ である。図２に示された命令フォーマットの説明において、３ビットの実行条件フィールド１０５を示したが、９個の条件を全て実現しようとすると、実は４ビットの実行条件フィールド１０５が必要である。しかし、実行条件フィールド１０５のビット長を長くして全ての条件をユーザに提供することよりも、条件を制限して実行条件フィールド１０５のビット長を短くし、その結果、演算フィールドに割り当てられるビット数を増加する方が、ユーザにとって便利であることも考えられる。

【００５９】そこで、既に説明したように、実行コントロールフラグＦ０，Ｆ１の数が２個の場合でも、３ビッ

トの実行条件フィールド１０５を定義した。図１１における実線で示された各組み合わせは、全組み合わせ９個のうち、破線で示された代用可能な組み合わせを除いた７種類である。ユーザは、破線で示された実行コントロールフラグＦ０＝偽、Ｆ１＝真の条件、および実行コントロールフラグＦ０＝偽、Ｆ１＝偽の条件を、他の条件、例えば、実行コントロールフラグＦ０＝真、Ｆ１＝偽の条件、および実行コントロールフラグＦ０＝真、Ｆ１＝真の条件で代用することになる。図２に示された命令フォーマットの説明において示された７種類の条件は、以上のような考え方にもとづいて定義されたものである。

【００６０】以上のように、実行条件の種類を制限すれば、演算フィールドに割り当てられるビット数をさらに増加することができる。

【００６１】実施の形態２．図１２はこの発明の実施の形態２によるマイクロプロセッサにおける命令デコードユニット２ｂにおいて条件実行を行うための構成を示すブロック図である。図において、４０４は命令中の３ビットの実行条件フィールド１０５の値にもとづいて該当レジスタを参照するレジスタ参照部、４０６は参照されたレジスタの設定にもとづいて実行条件を判定する実行条件判定部である。その他のものは図９に示したものと同じものである。

【００６２】次に動作について説明する。この場合、命令中の実行条件フィールド１０５は、以下のような意味を持つ。

コード：	実行条件
ＣＣ＝０００：	常時
００１：	Ｒ１で指定
０１０：	Ｒ２で指定
０１１：	Ｒ３で指定
１００：	Ｒ４で指定
１０１：	Ｒ５で指定
１１０：	Ｒ６で指定
１１１：	予約済

すなわち、実行条件フィールド１０５は、汎用レジスタ５を用いて間接的に実行条件を指定する。

【００６３】図１３はフラグの組み合わせ条件を指定するレジスタ２４０の内容を示す説明図である。レジスタ２４０中の各ビットによる実行条件は、図１８に示された条件と同じである。この場合、Ｃv0ビット２４１，Ｃd0ビット２４２は、実行コントロールフラグＦ０との比較に用いられるビットである。Ｃv1ビット２４３，Ｃd1ビット２４４は、実行コントロールフラグＦ１との比較に用いられるビットである。Ｃv2ビット２４５，Ｃd2ビット２４６は、実行コントロールフラグＦ２との比較に用いられるビットである。

【００６４】例えば、実行条件フィールド１０５に設定された値ＣＣ＝「００１」の場合には、レジスタ参照部４０４は、レジスタＲ１のＣv0ビット２４１，Ｃd0ビット２４２，Ｃv1ビット２４３，Ｃd1ビット２４４，Ｃv2ビット２４５，Ｃd2ビット２４６を入力する。そして、それを実行条件判定部４０６に出力する。実行条件判定部４０６は、Ｃv0ビット２４１およびＣd0ビット２４２による意味、Ｃv1ビット２４３およびＣd1ビット２４４による意味、Ｃv2ビット２４５およびＣd2ビット２４６による意味を、それぞれ、実行コントロールフラグＦ０の状態、実行コントロールフラグＦ１の状態、実行コントロールフラグＦ２の状態と比較する。各意味と各状態とが全て一致していたら、実行条件判定部４０６は、実行制御部４０３に対して、その命令を実行するように指示を出す。

【００６５】以上のように、この実施の形態２によれば、実行条件フィールド１０５に設定されたレジスタを指定する値に応じて該当レジスタを参照し、レジスタに設定されている値に応じて条件実行するので、やはり、実行条件フィールド１０５のビット数を低減することができ、演算フィールドに割り当てられるビット数を増加することができる。

【００６６】実施の形態３．図１４はこの発明の実施の形態３によるマイクロプロセッサにおける２演算命令のフォーマットを示す説明図である。図に示すように、このフォーマット２５０には、フィールド１０３とフィールド１０４とからなるフォーマットフィールド、２つの演算フィールド１０６，１０７、１つの実行条件フィールド１０５、および１つの命令選択条件フィールド２５１がある。図１５は実施の形態３によるマイクロプロセッサにおける命令デコードユニット２ｃにおいて条件実行を行うための構成を示すブロック図である。図において、４０５は命令選択条件フィールド２５１の内容と実行条件判定部４０２からの指示とに応じて命令の実行を制御する実行制御部である。その他のものは図９に示されたものと同じものである。

【００６７】次に動作について説明する。図１４に示すフォーマット２５０における命令選択条件フィールド２５１は、演算フィールド１０６によるoperation_0を実行するか演算フィールド１０７によるoperation_1を実行するか選択するための１ビットのＳビットを含む。Ｓビットは、以下のように用いられる。

ＣＣによる条件不成立のとき：Ｓビット無視（operation_0もoperation_1も実行しない）
ＣＣによる条件成立のとき　：Ｓビット＝０ならばoperation_0、operation_1ともに実行する

Ｓビット＝１ならばoperation_０を実行し、op
eration_１を実行しない

【００６８】実行条件デコード部４０１および実行条件判定部４０２は、実施の形態１の場合と同様に動作する。そして、実行条件フィールド１０５の値ＣＣによる条件が成立した場合には、実行制御部４０５に対して、命令を実行するに指示を与える。実行制御部４０５は、その指示を受けると、Ｓビットの判定を行う。そして、Ｓビット＝０ならば、メモリユニット３および整数演算ユニット４に制御信号１１，１２を出力する。Ｓビット＝１ならば、メモリユニット３に制御信号１１を出力する。

【００６９】この実施の形態３によれば、命令選択条件フィールド２５１によって、operation_０の実行およびoperation_１の実行を制御することができる。すなわち、２演算命令において、２つの演算を条件付きで実行することができる。なお、この実施の形態３では、実施の形態１に対してＳビットの判定処理が追加されたものを示したが、実施の形態２に対してＳビットの判定処理を追加してもよい。

【００７０】

【発明の効果】以上のように、請求項１記載の発明によれば、条件実行命令を有するマイクロプロセッサを、実行条件フィールドの値のデコード結果と汎用フラグによる条件とが合致しているか否か判定し、合致していた場合に命令を実行するように構成したので、実行条件フィールドに割り当てられるビット長を短くでき、その結果、演算フィールドに割り当てられるビット数を増加でき、マイクロプロセッサにおいて提供できる命令数を増やすことができる効果がある。

【００７１】請求項２記載の発明によれば、条件実行命令を有するマイクロプロセッサを、実行条件フィールドが条件実行の判定に用いられる汎用フラグの全ての組み合わせを表現するビット数よりも少ないビット長を有し、実行条件デコード部がそのようなビット長のエンコード値をデコードするように構成したので、実行条件フィールドに割り当てられるビット長をさらに短くできる効果がある。

【００７２】請求項３記載の発明によれば、条件実行命令を有するマイクロプロセッサを、実行条件フィールドで指定されたレジスタに設定されている条件と汎用フラグによる条件とが合致しているか否か判定し、合致していた場合に命令を実行するように構成したので、実行条件フィールドに割り当てられるビット長を短くでき、その結果、演算フィールドに割り当てられるビット数を増加でき、マイクロプロセッサにおいて提供できる命令数を増やすことができる効果がある。

【００７３】請求項４記載の発明によれば、条件実行命令を有するマイクロプロセッサを、命令中の命令選択条件フィールドの設定値に応じて複数の演算フィールドによる操作を行うか否か定めるように構成したので、２演算命令において、２つの演算を条件付きで実行することができる効果がある。

【図面の簡単な説明】

【図１】　この発明の実施の一形態によるマイクロプロセッサの構成を示すブロック図である。

【図２】　マイクロプロセッサの命令フォーマットを示す説明図である。

【図３】　演算フィールドの詳細な内容を示す説明図である。

【図４】　マイクロプロセッサのレジスタ構成を示す説明図である。

【図５】　ＰＳＷの詳細内容を示す説明図である。

【図６】　マイクロプロセッサの並列２命令実行時のパイプライン動作を示す説明図である。

【図７】　マイクロプロセッサのシーケンシャル命令実行時のパイプライン動作を示す説明図である。

【図８】　条件実行を用いるプログラムの一例を示す説明図である。

【図９】　この発明の実施の形態１における命令デコードユニットにおいて条件実行を行うための構成を示すブロック図である。

【図１０】　実行コントロールフラグを３個とした場合の実行条件を示す説明図である。

【図１１】　実行コントロールフラグが２個の場合の実行条件の設定の仕方を示す説明図である。

【図１２】　この発明の実施の形態２における命令デコードユニットにおいて条件実行を行うための構成を示すブロック図である。

【図１３】　フラグの組み合わせ条件を指定するレジスタの内容を示す説明図である。

【図１４】　この発明の実施の形態３によるマイクロプロセッサにおける２演算命令のフォーマットを示す説明図である。

【図１５】　この発明の実施の形態３における命令デコードユニットにおいて条件実行を行うための構成を示すブロック図である。

【図１６】　従来のＲＩＳＣマイクロプロセッサの命令フォーマットの一例を示す説明図である。

【図１７】　従来の他のマイクロプロセッサの命令フォーマットを示す説明図である。

【図１８】　Cv ビットおよびCd ビットとそれらが表す意味との関係を示す説明図である。

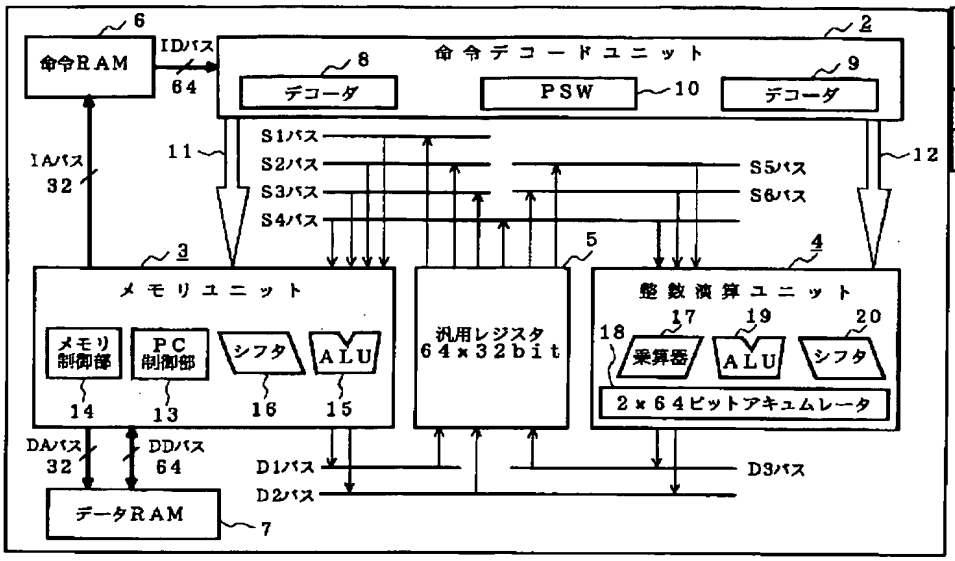
【符号の説明】

２　命令デコードユニット（命令デコーダ）、３　メモリユニット（命令実行部）、４　整数演算ユニット（命令実行部）、１５０　制御レジスタ、４０１実行条件デコード部、４０２，４０６　実行条件判定部、４０４

25

レジスタ参照部、４０５　実行制御部。

【図１】

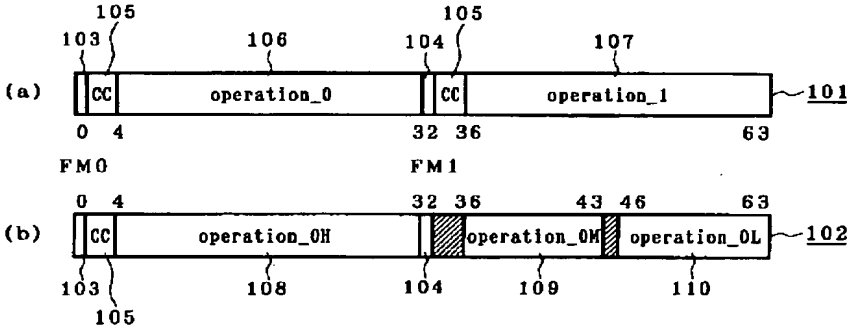


2:命令デコードユニット(命令デコーダ)　3:メモリユニット(命令実行部)　4:整数演算ユニット(命令実行部)

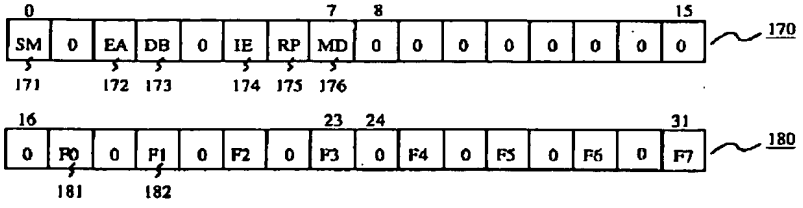
【図１８】

有効性Cv	値Cd	意味
1	1	真
1	0	偽
0	x	無視

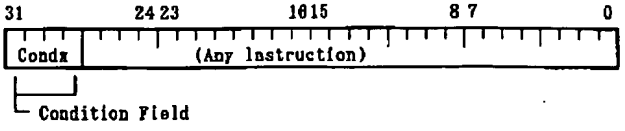
【図２】



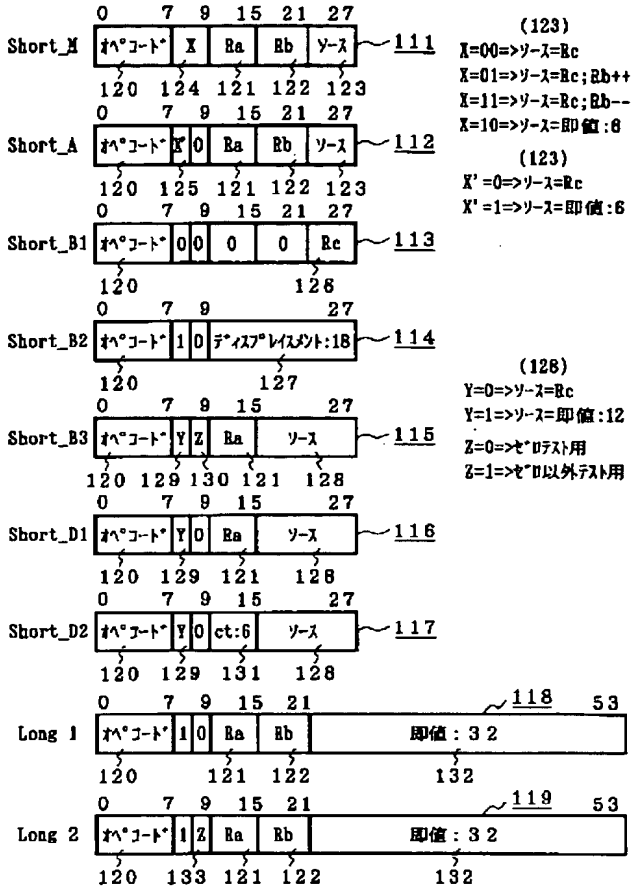
【図５】



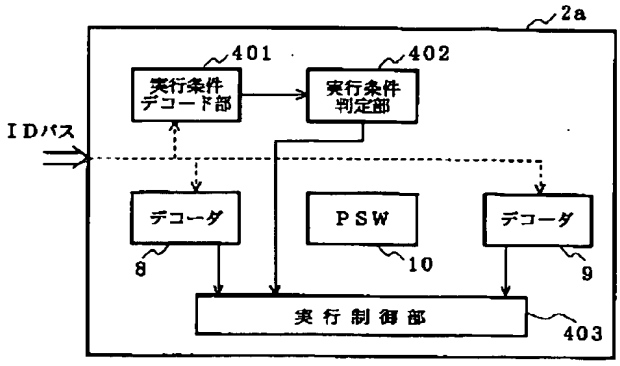
【図１６】



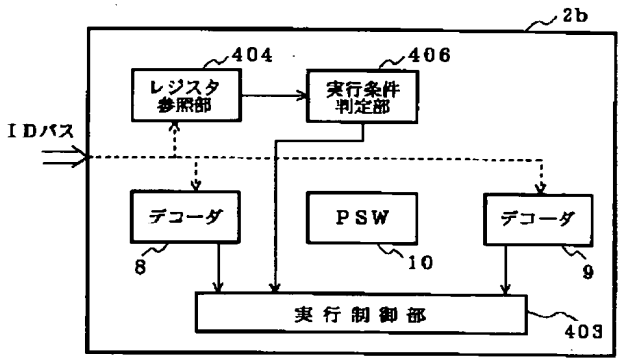
【図３】



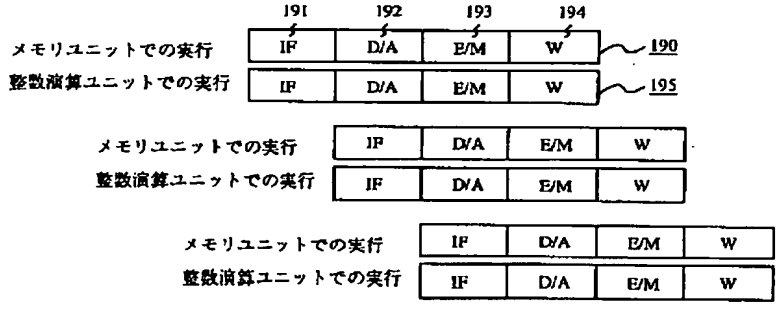
【図９】



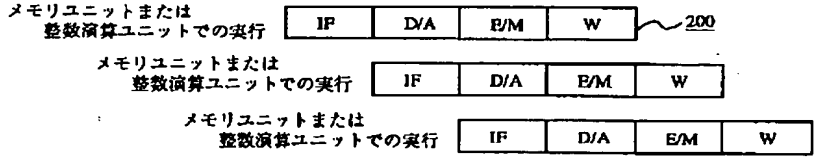
【図１２】



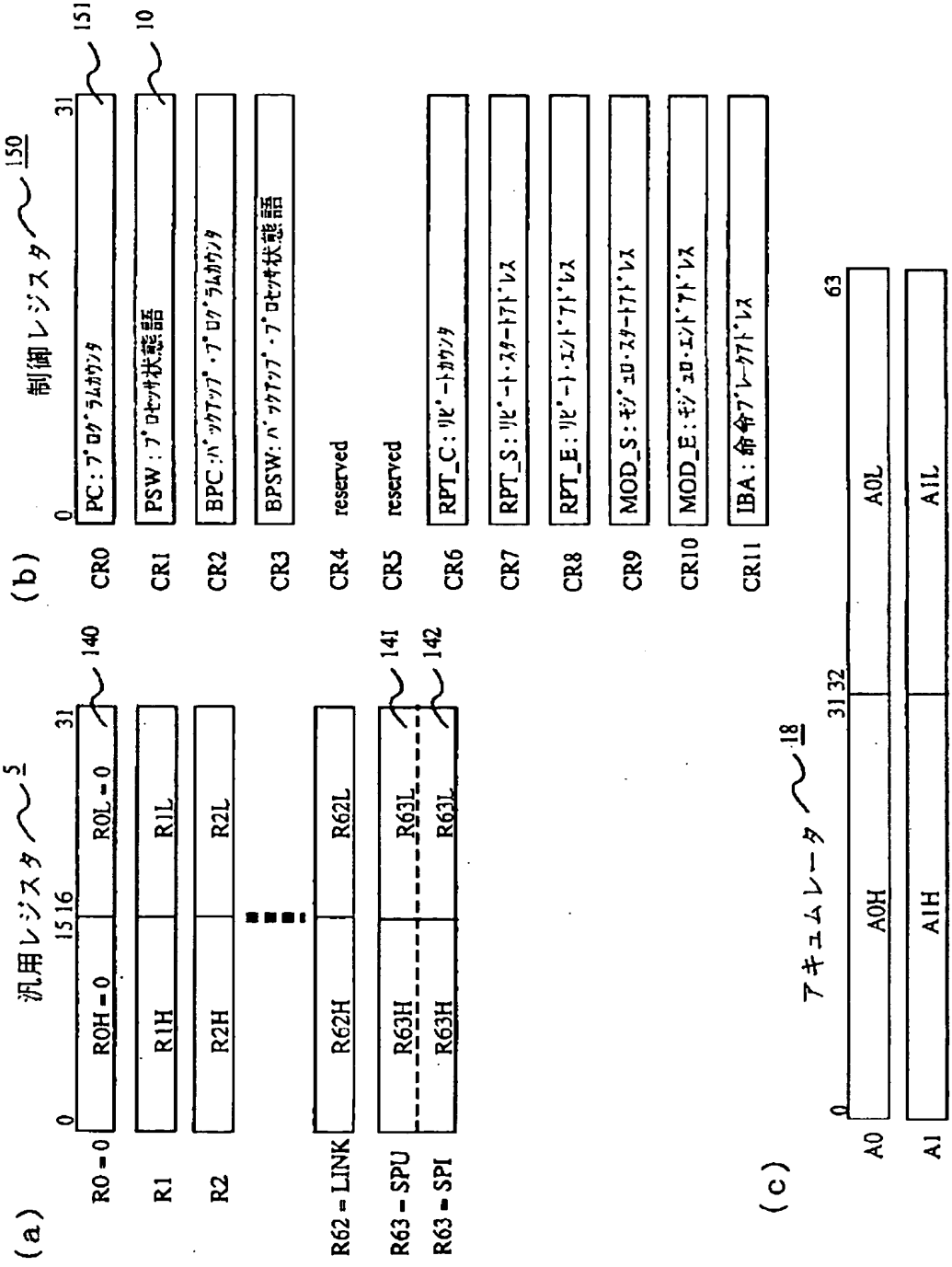
【図６】



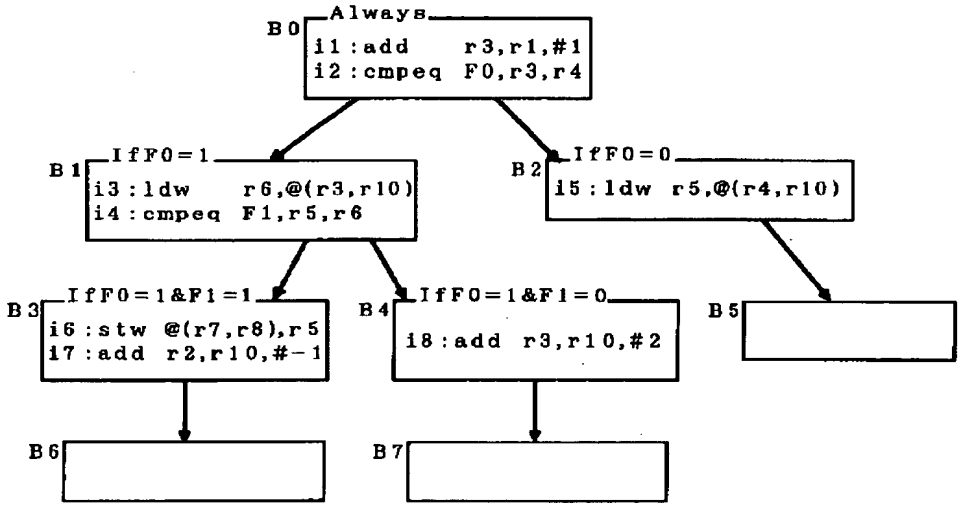
【図７】



【図４】


(a)
汎用レジスタ 5
140
31
15 16
0
R0H = 0
R0L = 0
R1H
R1L
R2H
R2L
R62H
R62L
R63H
R63L
141
142
R0 = 0
R1
R2
R62 = LINK
R63 = SPU
R63 = SPI
(b)
制御レジスタ 150
151
10
31
0
PC : プログラムカウンタ
PSW : プロセッサ状態語
BPC : バックアップ・プログラムカウンタ
BPSW : バックアップ・プロセッサ状態語
reserved
reserved
RPT_C : リピート・カウンタ
RPT_S : リピート・スタート・アドレス
RPT_E : リピート・エンド・アドレス
MOD_S : モジュロ・スタート・アドレス
MOD_E : モジュロ・エンド・アドレス
IBA : 命令プレーク・アドレス
CR0
CR1
CR2
CR3
CR4
CR5
CR6
CR7
CR8
CR9
CR10
CR11
(c)
アキュムレータ 18
63
31 32
0
A0H
A0L
A1H
A1L
A0
A1

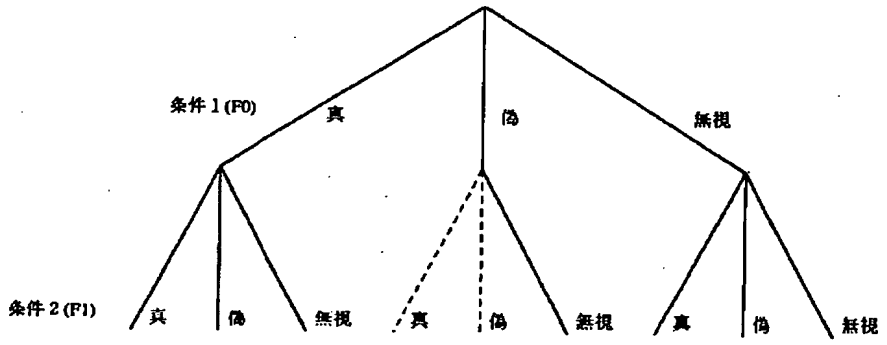
【図８】



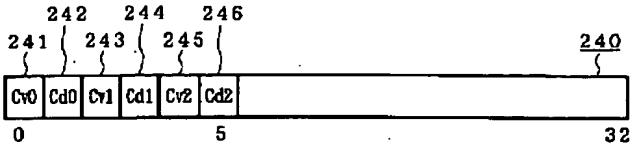
【図１０】

CC	F0	F1	F2	CC	F0	F1	F2	CC	F0	F1	F2
00000	真	真	真	01001	偽	真	真	10010	無視	真	真
00001	真	真	偽	01010	偽	真	偽	10011	無視	真	偽
00010	真	真	無視	01011	偽	真	無視	10100	無視	真	無視
00011	真	偽	真	01100	偽	偽	真	10101	無視	偽	真
00100	真	偽	偽	01101	偽	偽	偽	10110	無視	偽	偽
00101	真	偽	無視	01110	偽	偽	無視	10111	無視	偽	無視
00110	真	無視	真	01111	偽	無視	真	11000	無視	無視	真
00111	真	無視	偽	10000	偽	無視	偽	11001	無視	無視	偽
01000	真	無視	無視	10001	偽	無視	無視	11010	無視	無視	無視

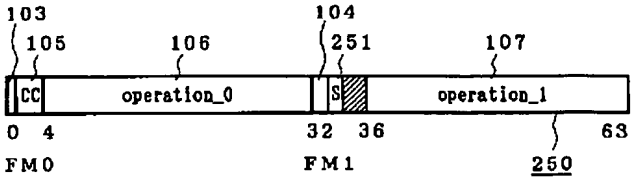
【図１１】



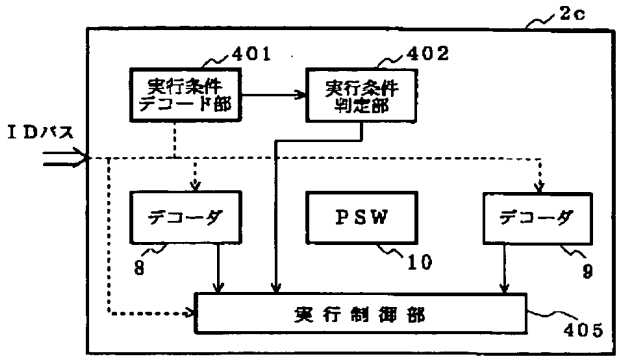
【図１３】



【図１４】



【図１５】



【図１７】